## **DAILY ONLINE ACTIVITIES SUMMARY**

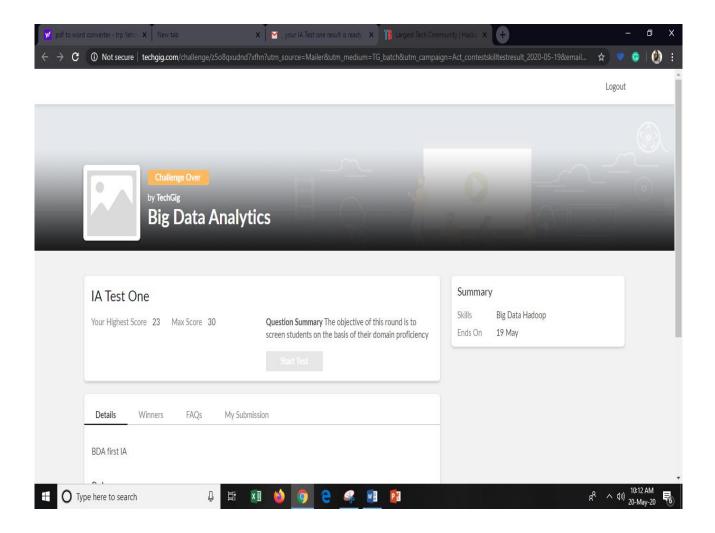
Date:	19-05-2020		Name: Shetty		Sonali Sanjeeva
Sem & Sec	8 <sup>th</sup> , B		USN:	4AL16	CCS123
Online Test Summary					
Subject Big data analytic					
Max. Marks	30		Score 23		
Certification Course Summary					
Course DEEP LEARNING ONRAMP BY MATHWORKS					
Certificate Provider		ICT academy	Duration		2 Hrs
Coding Challenges					
Problem Statement: prob1- To add some letters for a given word or letter then to find the shortest palindrome possible  Prob2- To check whether the given linked list is palindrome or not					
Status: Solved					
Uploaded the report in Github			Yes		
If yes Repository name			SONALI SHETTY		
Uploaded the report in slack			Yes		

Online Test Details: (Attach the snapshot and briefly write the report for the same

Certification Course Details: (Attach the snapshot and briefly write the report for the same)

Coding Challenges Details: (Attach the snapshot and briefly write the report for the same)

### 1) Online Test Details:



## 2) Certification Course Details:



# **Course Completion Certificate**

Shetty Sonali

has successfully completed 100% of the self-paced training course

Deep Learning Onramp

DIRECTOR, TRAINING SERVICES

06 May 2020

## 3) Coding Challenges:

- 1. We have a Letter or a word then we need add some letters to it and need to find out shortest palindrome example we take "S": S will be the shortest palindrome string. If we take "xyz": zyxyz will be the shortest palindrome string

  So we need to add some characters to the given string or character and find out what will be the shortest palindrome string by using simple java program.
- 2. Write a simple code to identify given linked list is palindrome or not by using stack. First take a Stack. Traverse through each node of the linked list and push each node value to Stack. Once the traversal & copying is done, iterate through linked list from head node again. In each iteration, pop one stack element and compare with node value in respective iteration. It is expected to match stack popped value with node value. In case of all matches, its a palindrome. Any one element mismatch makes it not a palindrome.

#### PROGRAM1

package shortestpalindromeexample.java; import java.util.Scanner; public class ShortestPalindromeDemo { public static String shortestPalindrome(String str) { int x=0;

```
int y=str.length()-1;
while(y>=0){
if(str.charAt(x)==str.charAt(y)){
       X++;
}
y--;
if(x==str.length())
return str;
String suffix = str.substring(x);
String prefix = new StringBuilder(suffix).reverse().toString();
String mid = shortestPalindrome(str.substring(0, x));
return prefix+mid+suffix;
}
public static void main(String[] args) {
Scanner in = new Scanner(System.in);
System.out.println("Enter a String to find out shortest palindrome");
String str=in.nextLine();
System.out.println("Shortest palindrome of "+str+" is "+shortestPalindrome(str));
}
PROGRAM 2
import java.util.Stack;
// Data Structure to store a linked list node
class Node {
       int data;
       Node next:
Node(int i)
this.data = i;
this.next = null;
}
};
class Main
// Function to determine if a given linked list is palindrome or not
public static boolean isPalindrome(Node head)
// construct an empty stack
Stack s = new Stack<>();
// push all elements of the linked list into the stack
```

```
Node node = head;
while (node != null) {
       s.push(node.data);
       node = node.next;
}
// traverse the linked list again
node = head;
while (node != null)
// pop the top element from the stack
int top = s.pop();
// compare the popped element with current node's data
// return false if mismatch happens
if (top != node.data) {
return false;
// advance to the next node
node = node.next;
}
// we reach here only when the linked list is palindrome
return true;
public static void main(String[] args)
       Node head = new Node(1);
       head.next = new Node(2);
       head.next.next = new Node(3);
       head.next.next.next = new Node(2);
       head.next.next.next.next = new Node(1);
if (isPalindrome(head)) {
       System.out.print("Linked List is a palindrome.");
} else {
       System.out.print("Linked List is not a palindrome.");
}
}
```